`conjacp` R function for paired conjoint analyses

Flavien Ganter

October 30, 2020

1 Description

`conjacp` is a function designed to analyze data from paired conjoint experiments. Specifically, it estimates average component preferences, differences in average component preferences between subgroups, and direct pairwise preferences (Ganter 2020).

2 Usage

```
conjacp(formula, data, tasks, estimand = "acp", subgroups = NULL,
by = NULL, adjust = FALSE, id = NULL, level_weights = NULL, condition
= NULL, subset = NULL)
```

3 Arguments

`formula` A formula specifying an ACP model to be estimated. Two-way restrictions in the experimental design can be specified as interactions; higher order constraints are not allowed. The outcome variable should be a numeric or integer dummy variable that takes values 1 and 0. Covariates can be numeric or integer (continuous attributes) or factor (categorical attributes); variables involved in two-way constraints can only be of class factor.

`data` A data frame containing variables specified in `formula` as well as additional variables used to cluster the analysis, or to compare subgroups. The data has to be "long" (one observation per profile, that is, two observations per task).

`tasks` A named vector that identifies pairs of profiles that were simultaneously presented to respondents.

`estimand` A character string specifying the quantity of interest to be estimated. Options are average component preferences (`"acp"`, default), differences in average component preferences (`"dacp"`), and direct pairwise preferences (`"p"`).

`subgroups` An (optional) named vector identifying factor variables that split the sample into subgroups of respondents, to estimate the quantities of interest for subsets of respondents (see `subset` argument).

`by` An (optional) character string identifying a two-level factor variable that specifies the two subgroups of respondents the differences in ACPs should be calculated on.

`adjust` A logical value indicating whether values taken by other attributes should be included in models estimated to calculate the quantities of interest for the focal attribute. When `adjust = FALSE`, the vector $\Gamma V_{i[-\ell]}$ is omitted (Ganter 2020: 22).

`id` An (optional) character string identifying respondents, to be used for clustering standard errors.

`level_weights` An (optional) list specifying the relative weight associated with each level. The list should contain one sublist per attribute; the first element of the sublist has to be the attribute name (same as the one used in the formula and in the data), and the second element is a vector of weights. By default, all levels are identically weighted. Level weights are automatically normalized by the function so that they always sum to 1.

`condition` An (optional) two-element list specifying the attribute and the level of that attribute the estimated ACPs should be conditional on.

`subset` An (optional) two-element list specifying a subgroup factor and a level of that subgroup factor, to calculate ACPs on a subset of respondents. Requires that the `subgroups` argument be specified accordingly.

4 Value

`conjacp` returns an list of class "conjacp" that can be accessed through the command `summary` and that contains (among other things, used by the `print` and `summary` commands):

`estimates` The vector of coefficients; ACPs, DCSPs, and direct pairwise preferences. For attributes that are restricted (conditionally independent randomization), ACPs conditional on unrestricted levels only are listed.

`vcov` The variance-covariance matrix associated with the vector `estimates` that can be used to obtain standard errors for coefficients or for transformations of the coefficients.

`estimates_alt` The vector of coefficients; ACPs, DCSPs, and direct pairwise preferences. For attributes that are restricted (conditionally independent randomization), ACPs conditional on "comparable pairs" only are listed. The levels of the focal attribute for both profiles are both either restricted or unrestricted; or, they are mixed and the levels of the complementary attribute are unrestricted for both profiles.

`vcov_alt` The variance-covariance matrix associated with the vector `estimates_alt` that can be used to obtain standard errors for coefficients or for transformations of the coefficients.

The `print` and `summary` commands return unconditional quantities of interest (for unrestricted attributes) and quantities of interest conditional on unrestricted pairs (for restricted attributes). To obtain quantities of interest conditional on "comparable pairs," the argument `cacp = "comparable_pairs"` should be specified. Except for differences in ACPs, the command only returns coefficients and standard errors, not z- and p-values, for testing the nullity of the ACPs and direct pairwise preferences has no interest.

5 Additional functions

The `conjacp` function brings together two sub-functions: `conjacp.prepdata`, which prepares the data for the analysis, and `conjacp.estimation`, which performs the estimation. If several estimations needs to be performed on the same data (for different estimands, conditions, or subsets of respondents), it is possible to run the former only once and to use its output as the input for the iterations of the latter. The `conjacp.prepdata` function takes the arguments `formula`, `data`, `tasks`, `subgroups`, `id`, and `level_weights`; the `conjacp.estimation` function takes the arguments `conjacp.prepdata.object` (output from `conjacp.prepdata`), `estimand`, `by`, `adjust`, `condition`, and `subset`.

The `conjacp` function additionally comes with an additional command `conjacp_diff`, designed to performed pairwise comparisons between any pair of quantities estimated, and that provides z- and p-values for the differences. It takes four arguments:

`object` A `conjacp` object.

`acp1,acp2` Two named arguments that specify the quantities to be compared. The names should exactly match the names displayed in the coefficient table provided by the command `summary`.

`cacp` A character string specifying if, for restricted attributes, the comparison should involve ACPs conditional on unrestricted levels (`"conditional"`, default) or conditional on "comparable pairs" (`"comparable_pairs"`).

The `conjacp` function also comes with a `conjacp.var` command that calculates the range and variability of ACPs for each attribute (see supplemental information B; Ganter 2020). It takes four arguments:

`object` A `conjacp` object.

`cacp` A character string specifying if, for restricted attributes, the range et variability statistics should involve ACPs conditional on unrestricted levels (`"conditional"`, default) or conditional on "comparable pairs" (`"comparable_pairs"`).

`alpha` An (optional) numeric argument in the]0;1[interval specifying the level of the confidence intervals. Default is .05.

`nsimul` An (optional) positive integer specifying the number of simulations used to estimate confidence intervals. Default is 1,000.

The function returns the estimates of the range and variability for all attributes, along with their variance-covariance matrix, and the upper and lower bounds of the confidence intervals for the estimates.

References

Ganter, Flavien. 2020. "Identification of Preferences in Forced-Choice Conjoint Experiments: Reassessing the Quantity of Interest." SocArXiv. October 30. doi:10.31235/osf.io/e638u.